

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів

Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Рудницька Єлизавета Сергіївна

(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.92:54

КВАЛІФІКАЦІЙНА РОБОТА

CRM-система для управління взаємодією з клієнтами ветеринарного центру

(тема роботи)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Гіваргізов Інвія Геннадійович

(прізвище, ім'я, по батькові)

старший викладач КТіМС, к. е. н.

(науковий ступінь, вчене звання)

Житомир – 2026

Висновок кафедри комп'ютерних технологій і моделювання систем
за результатами попереднього захисту: допущена

Протокол засідання кафедри комп'ютерних технологій і моделювання систем

№ 20 від « 05 » червня 2026р.

Завідувач кафедри комп'ютерних технологій і моделювання систем

к. пед. н.. доцент

(науковий ступінь, вчене звання)

(підпис)

КОВАЛЬЧУК М. О.

(прізвище, ім'я, по батькові)

« 05 » червня 2026 р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище ,ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

лаборант кафедри

(науковий ступінь, вчене звання)

(підпис)

КОРОЛЬЧУК В.В.

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Рудницька Є. С. «CRM-система для управління взаємодією з клієнтами ветеринарного центру». – Кваліфікаційна робота на правах рукопису. 56 с., 12 рис., 6 табл., 45 джерел, 4 додатки.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю **122 – комп'ютерні науки**. – Поліський національний університет, Житомир, 2026.

Об'єкт дослідження – процес управління взаємодією з клієнтами у ветеринарному центрі.

Мета роботи – розроблення CRM-системи, що забезпечує автоматизацію бізнес-процесів ветеринарного центру, включаючи облік клієнтів і тварин, планування прийомів, ведення медичної документації, управління послугами та фінансовими операціями.

Методи дослідження – аналіз і синтез, моделювання інформаційних процесів, побудова ER- та DFD-діаграм, об'єктно-орієнтоване проєктування, методи програмної інженерії та емпіричні методи тестування.

У кваліфікаційній роботі виконано аналіз бізнес-процесів ветеринарного центру та огляд сучасних CRM-систем медичного й ветеринарного напрямів. На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до розроблюваної системи. Побудовано інфологічну модель даних, структурні та функціональні діаграми, спроектовано архітектуру CRM-системи на основі трирівневого підходу. Реалізовано основні програмні модулі, включно з обліком клієнтів і пацієнтів, календарем прийомів, електронною медичною карткою, фінансовим обліком та засобами адміністрування. Проведено тестування ключових сценаріїв і підготовлено інструкцію користувача. Розроблена система забезпечує автоматизацію основних бізнес-процесів ветеринарного центру, підвищує ефективність роботи персоналу, якість обслуговування клієнтів і точність ведення медичної документації.

Ключові слова: CRM-система, ветеринарний центр, база даних, інформаційна система, трирівнева архітектура.

SUMMARY

Rudnytska Ye. S. “CRM System for Managing Client Interaction in a Veterinary Center.” – Qualification paper in manuscript form. 56 pages, 12 figures, 6 tables, 45 references, 4 appendices.

Qualification paper for obtaining the bachelor’s degree in specialty **F3 – Computer Science**. – Polissia National University, Zhytomyr, 2026.

Object of the study – the process of managing client interaction in a veterinary center.

Aim of the study – development of a CRM system that automates the business processes of a veterinary center, including client and animal record management, appointment scheduling, medical documentation, service management, and financial operations.

Research methods – analysis and synthesis, modelling of information processes, construction of ER and DFD diagrams, object-oriented design, software engineering methods, and empirical testing methods.

The qualification paper presents an analysis of the business processes of a veterinary center and a review of modern CRM systems in the medical and veterinary domains. Based on this analysis, functional and non-functional requirements for the developed system were formulated. An infological data model, structural and functional diagrams were created, and a three-tier architecture for the CRM system was designed. The main software modules were implemented, including client and patient management, appointment scheduling, electronic medical records, financial accounting, and administration tools. Key scenarios were tested, and a user manual was prepared. The developed system ensures automation of the core business processes of a veterinary center, improving staff efficiency, service quality, and the accuracy of medical documentation.

Keywords: CRM system, veterinary center, database, information system, automation, electronic medical record, REST API, three-tier architecture.

ВСТУП

Сучасний розвиток цифрових технологій та зростання обсягів інформації, що обробляється у сферах медичних та сервісних послуг, зумовлюють потребу у впровадженні ефективних інформаційних систем управління взаємодією з клієнтами. Ветеринарні центри, як і медичні установи для людей, працюють із великим масивом даних: інформацією про клієнтів та їхніх тварин, історією візитів, записами на прийом, послугами, фінансовими операціями та внутрішніми процесами. Традиційні підходи до ведення обліку, які базуються на паперових картках або розрізнених електронних документах, є малоефективними, призводять до дублювання інформації, втрати даних, помилок та значних витрат часу персоналу.

Тому актуальним є застосування CRM-систем – програмних рішень, що забезпечують автоматизацію бізнес-процесів, підвищення якості сервісу, створення повної історії взаємодії з клієнтами, оптимізацію роботи персоналу й управління ветеринарними послугами. Використання спеціалізованої CRM-системи дозволяє централізувати всі дані, забезпечити швидкий доступ до них, знизити навантаження на працівників реєстратури та ветеринарів, мінімізувати помилки під час прийому, покращити комунікацію з власниками тварин та забезпечити якісну підтримку бізнес-процесів ветеринарного центру.

Метою роботи є розроблення CRM-системи для управління взаємодією з клієнтами ветеринарного центру, яка забезпечує автоматизацію процесів обліку клієнтів і тварин, формування записів на прийом, ведення історії візитів, управління переліком послуг та підтримку внутрішньої роботи персоналу.

Для досягнення поставленої мети у роботі вирішуються такі **завдання**:

- проаналізувати предметну область діяльності ветеринарного центру та визначити ключові інформаційні процеси;
- провести огляд існуючих аналогів та визначити їхні недоліки;
- сформулювати вимоги до функціоналу розроблюваної CRM-системи;
- розробити інфологічну модель бази даних та структурні схеми функціонування системи;

- спроектувати архітектуру та REST-сервіси;
- реалізувати основні програмні модулі CRM-системи;
- розробити інтерфейс користувача відповідно до сучасних вимог usability;
- виконати тестування роботи системи та підготувати інструкцію користувача.

Об’єктом дослідження є процес управління взаємодією з клієнтами у ветеринарному центрі.

Предметом дослідження є моделі, методи та програмні засоби побудови CRM-системи для автоматизації роботи ветеринарного центру.

Методи дослідження. У роботі застосовано:

- методи аналізу та синтезу – для формування вимог і побудови функціональної структури системи;
- методи моделювання – для створення DFD-діаграм, UML-діаграм та інфологічної моделі бази даних;
- об’єктно-орієнтований підхід – для проектування програмних модулів;
- методи програмної інженерії – для реалізації архітектури та функціональних компонентів;
- емпіричні методи – для тестування на основі реальних сценаріїв використання.

Практичне значення отриманих результатів полягає у створенні працюючої CRM-системи, яка може бути впроваджена у ветеринарні клініки та центри для оптимізації бізнес-процесів, підвищення якості обслуговування клієнтів та покращення контролю над внутрішньою діяльністю персоналу. Система забезпечує швидкісний доступ до медичних даних тварин, автоматизацію записів, формування довідкової та фінансової документації.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг роботи становить 48 сторінок друкованого тексту, містить рисунки, таблиці, графічні матеріали, фрагменти коду та допоміжні документи згідно зі структурними вимогами методичних рекомендацій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Особливості діяльності ветеринарного центру та аналіз бізнес-процесів

Діяльність ветеринарного центру характеризується багатокomпонентністю та високим рівнем відповідальності, оскільки основні процеси пов'язані з наданням медичних послуг тваринам, консультуванням власників та забезпеченням належної якості сервісу. Робота закладу включає як медичні процедури, так і низку адміністративних та організаційних операцій, що вимагають точного й оперативного обліку даних. Саме тому інформаційні системи, що підтримують такі процеси, повинні забезпечувати комплексне управління взаємодією з клієнтами та пов'язаними об'єктами обліку.

У типовому ветеринарному центрі ключовими є такі бізнес-процеси:

1. Прийом та реєстрація клієнтів.

Кожен клієнт є власником однієї або кількох тварин, що зумовлює необхідність ведення детальної контактної інформації, зберігання історії взаємодій, фіксації звернень та нагадувань. Процес реєстрації включає первинний збір даних, уточнення інформації під час наступних звернень, створення профілю клієнта та підтримку актуальності записів.

2. Облік пацієнтів – домашніх тварин.

Для кожної тварини формується окремий запис з даними щодо віку, породи, фізіологічних особливостей, щеплень, хронічних захворювань та інших характеристик, що важливі для якісної діагностики. Правильність та повнота цих даних впливають на точність медичних рішень ветеринара.

3. Управління записом на прийом.

Процес запису включає календарне планування, вибір лікаря, визначення часу та типу послуги. У реальних умовах виникають задачі оптимізації розкладу, уникнення конфліктів та дублювань, перенесення прийомів і формування нагадувань для клієнтів. Важливо, щоб система автоматично повідомляла про зміни, враховувала зайнятість ветеринарів і забезпечувала зручність для персоналу.

4. Проведення медичного огляду та фіксація результатів.

Під час візиту лікар створює медичний запис: опис стану тварини, діагноз, призначення ліків, перелік проведених процедур та рекомендації для подальшого лікування. Формування єдиної цифрової історії відвідувань дозволяє уникнути втрати важливих даних та забезпечує безперервність лікування.

5. Облік і надання послуг.

Ветеринарні центри пропонують широкий спектр послуг: консультації, щеплення, лабораторні аналізи, хірургічні втручання, профілактичні процедури тощо. Кожна послуга має свій тариф, тривалість виконання, вимоги до обладнання та медикаментів. Ефективне управління переліком послуг дозволяє формувати точні розрахунки та уникати помилок у фінансових операціях.

6. Фінансовий облік та розрахунки.

Фінансовий процес включає формування рахунків, облік оплат, контроль заборгованостей та створення фінансових звітів. Важливо забезпечити прозорість усіх транзакцій, можливість аналізу прибутковості послуг, відстеження витрат та правильне документування операцій.

7. Комунікація з клієнтами.

Сюди входять нагадування про візити, повідомлення про вакцинацію, інформування про результати лабораторних досліджень, надсилання рекомендацій та новин. Чітка та своєчасна комунікація підвищує рівень сервісу і сприяє формуванню довіри клієнтів.

8. Внутрішні процеси та адміністрування персоналу.

Адміністратор або керівництво центру контролюють робочі графіки ветеринарів, розподіл завдань, ведення внутрішньої документації та моніторинг навантаження персоналу. Для цього необхідні інструменти контролю доступу, авторизації та ведення журналів дій.

У сукупності ці процеси формують складну інформаційну систему взаємозалежних даних. Неефективність хоча б одного з них призводить до помилок, затримок у роботі, пропущених прийомів та погіршення якості обслуговування. Тому впровадження CRM-системи є логічним і необхідним етапом підвищення ефективності роботи ветеринарного центру.

CRM-система дозволяє автоматизувати операції, зменшити навантаження на персонал, забезпечити швидкий доступ до даних, підвищити продуктивність та оптимізувати взаємодію між клієнтом і ветеринаром. Аналіз бізнес-процесів демонструє, що така система повинна забезпечувати централізацію інформації, підтримку календарного планування, ведення медичних записів, облік фінансів, управління персоналом і комунікацію з клієнтами в єдиному цифровому середовищі.

1.2 Огляд існуючих CRM-систем для медичних і ветеринарних закладів

Сегмент CRM-рішень (Customer Relationship Management – управління взаємовідносинами з клієнтами) для медичних і ветеринарних закладів демонструє активний розвиток. Сучасні клініки гостро потребують комплексної автоматизації бізнес-процесів, підвищення якості обслуговування клієнтів та забезпечення прозорості внутрішньої діяльності. Ринок пропонує як універсальні CRM-платформи, що можуть бути адаптовані під галузь, так і спеціалізовані медичні інформаційні системи, створені безпосередньо для клінічної діяльності.

Розглянемо детальніше ці системи, щоб визначити їхні основні функціональні можливості, типові недоліки та вимоги, які слід врахувати при розробці власної CRM-системи.

CRM-системи загального призначення

До цієї групи належать універсальні програмні рішення, які використовуються в різних галузях бізнесу, включно з медициною. Вони пропонують стандартизовані інструменти для роботи з клієнтами (база контактів, комунікації, воронка продажів), але їх використання у ветеринарних центрах часто вимагає суттєвої та дороговартісної адаптації.

1. Salesforce Health Cloud – це потужне розширення корпоративної CRM-платформи Salesforce, орієнтоване на медичні заклади.

Ключові можливості: підтримка роботи з пацієнтами, комплексне планування прийомів, багатоканальна комунікація та розширена аналітика.

Переваги: висока масштабованість, можливість глибокої кастомізації (налаштування під потреби користувача), надійність.

Недоліки: впровадження є дорогим і складним, що робить цю систему недоступною для невеликих ветеринарних клінік.

2. *Zoho CRM + Zoho Clinic Extensions* – є модульною системою, яка використовується у великій кількості галузей.

Ключові можливості: модульна структура дозволяє додавати медичні розширення (*Clinic Extensions*) для ведення карток пацієнтів та запису на прийом.

Переваги: гнучкість базової CRM, доступність для малого бізнесу.

Недоліки: обмежена гнучкість спеціалізованих медичних функцій. Відсутність тісної інтеграції з унікальними ветеринарними процесами (наприклад, ведення даних про тварин, вакцинацію, фізіологічні особливості та історію щеплень).

Отже, універсальні CRM-системи містять широкий набір інструментів, але не завжди враховують критичну специфіку ветеринарного обслуговування. Щоб вирішити цю проблему, існують спеціалізовані рішення.

Спеціалізовані ветеринарні CRM та медичні інформаційні системи

Ці системи створювалися саме для ветеринарних клінік та медичних установ, тому вони враховують галузеву специфіку та підтримують ключові бізнес-процеси без необхідності значної доробок.

1. *VetCRM* – система орієнтована на середні ветеринарні центри та клініки.

Основний функціонал: ведення клієнтів і тварин; облік медичних карток та історії візитів; планування прийомів та календар лікарів; управління послугами та оплатами.

Переваги: зручний інтерфейс, пряма адаптація до ветеринарної сфери.

Недоліки: обмежені можливості інтеграції із зовнішніми сервісами та нестача розширених інструментів аналітики.

2. *Provet Cloud* – хмарна ветеринарна система, що пропонує розширений функціонал для великих клінік.

Основний функціонал: інтеграція з лабораторним обладнанням; управління

інвентарем (запасами медикаментів); розширене медичне документування; мобільний доступ.

Переваги: широкі можливості для масштабування та комплексного управління.

Недоліки: висока вартість ліцензій та відносна складність інтерфейсу, що ускладнює роботу для малого бізнесу.

3. *eVetPractice (IDEXX)* – це комплексна система, часто інтегрована з діагностичними рішеннями IDEXX.

Основний функціонал: електронні медичні записи (ehr); планування та автоматичні нагадування клієнтам; формування фінансових звітів; інтеграція з діагностичним обладнанням.

Переваги: глибока медична та діагностична інтеграція.

Недоліки: обмеження для невеликих закладів, складна процедура налаштування.

4. *SimpleVET* – просте та доступне рішення, спеціально розроблене для невеликих ветеринарних практик.

основний функціонал: електронні медичні картки; календар записів; облік клієнтів та рахунків.

Переваги: простота використання, низька вартість.

Недоліки: мінімальний набір функцій, слабка масштабованість.

Окрім спеціалізованих ветеринарних систем, існують також *медичні інформаційні системи* (МІС) загального призначення, які домінують у сфері людської медицини.

Медичні інформаційні системи загального призначення (МІС)

Ці системи є найбільш поширеними у сфері медицини і часто використовуються в медичних закладах України, хоча й не мають прямого функціоналу для роботи з тваринами.

1. *Medstar / Helsi / Doctor Eleks* – це популярні в Україні МІС-системи, які забезпечують високий рівень автоматизації та взаємодії з державними реєстрами.

Основний функціонал: ведення електронних медичних карток пацієнтів-

людей; електронні записи та планування прийомів; фінансовий та статистичний облік; взаємодія з державними реєстрами (що критично для людської медицини).

Переваги: висока автоматизація та повна відповідність регуляторним вимогам України.

Недоліки: повна відсутність функцій для роботи з тваринами та специфічними ветеринарними процесами.

На основі детального огляду ринку можна сформулювати загальні тенденції та зробити висновки щодо доцільності розробки нової системи:

- Автоматизація ключових процесів: спрощення запису, реєстрації та документообігу.
- Централізація даних: зберігання всієї інформації про клієнтів та пацієнтів (тварин) в одному місці.
- Підтримка календаря лікарів: ефективне управління робочим часом персоналу.
- Ведення медичних записів: створення електронної історії хвороби.
- Покращення комунікації: автоматичні нагадування та оповіщення для клієнтів.
- Електронні фінансові операції: облік рахунків та оплат.

Характерні недоліки більшості рішень:

- Складність адаптації: універсальні системи важко і дорого налаштувати під потреби ветеринарної клініки.
- Проблеми з локалізацією: відсутність або обмежена підтримка української мови та локальних вимог.
- Висока вартість: висока ціна ліцензій та впровадження, особливо для малих і середніх закладів.
- Складні інтерфейси: часто системи не орієнтовані на малий та середній бізнес, мають надмірно складний функціонал.
- Недостатня підтримка ветеринарної специфіки: неповне або незручне ведення специфічних ветеринарних процесів (детальні записи тварин з фізіологічними параметрами, повна історія щеплень, паспорти).

Аналіз сучасних CRM-систем демонструє, що, хоча на ринку існують рішення для медичної та ветеринарної сфер, жодна система не пропонує оптимального поєднання функціональності, локалізації, доступності та простоти використання, особливо для малих і середніх ветеринарних центрів. Це підтверджує актуальність розроблення спеціалізованої CRM-системи, яка враховує унікальні особливості ветеринарної галузі та забезпечує комплексну підтримку всіх бізнес-процесів: від реєстрації клієнтів до ведення медичних записів і управління фінансовими операціями.

1.3 Постановка задачі та формування функціональних вимог до системи

Ефективне управління діяльністю ветеринарного центру передбачає автоматизацію комплексного набору бізнес-процесів, що охоплюють взаємодію з клієнтами, ведення обліку тварин, планування прийомів, формування медичних записів, управління послугами та фінансовими операціями. Традиційні методи, засновані на використанні паперових журналів або розрізнених електронних документів, не забезпечують належної точності, швидкості обробки даних та узгодженості інформації між різними підрозділами. Це призводить до зниження якості обслуговування та ускладнює оперативне управління клінікою.

У зв'язку з цим виникає необхідність розроблення CRM-системи, яка дозволить об'єднати всі ключові процеси в єдиному цифровому середовищі, забезпечити централізоване управління інформацією та створити інструменти для ефективної взаємодії між персоналом, клієнтами та базою даних ветеринарної клініки.

Потрібно розробити CRM-систему для ветеринарного центру, яка забезпечить автоматизацію процесів обліку клієнтів і тварин, управління записами на прийом, ведення історії візитів, адміністрування послуг, контроль фінансових операцій, комунікацію з клієнтами та підтримку роботи персоналу.

Система має забезпечувати зручний інтерфейс для працівників різних категорій (адміністраторів, ветеринарів, реєстраторів), підтримувати

розмежування доступу, інтегруватися з базою даних та забезпечувати високий рівень захисту інформації.

Основна задача полягає у створенні програмного продукту, що дозволить підвищити ефективність роботи ветеринарного центру шляхом усунення дублювання інформації, скорочення часу обробки даних, зменшення кількості помилок та підвищення якості сервісу для клієнтів.

Функціональні вимоги формуються відповідно до бізнес-процесів, описаних у попередніх підрозділах, та враховують потреби персоналу й клієнтів ветеринарного центру.

1. Реєстрація та управління клієнтами

Система повинна забезпечувати: створення та редагування картки клієнта з базовими персональними даними; ведення контактної інформації (телефон, email, адреса); пошук клієнтів за різними параметрами; фіксацію історії звернень та взаємодій; можливість додавання внутрішніх коментарів та нотаток.

2. Ведення обліку тварин

Функції системи повинні включати: створення медичної картки тварини із зазначенням виду, породи, статі, дати народження, ідентифікаційних параметрів; ведення інформації про попередні захворювання, алергії, вакцинації; прив'язку тварини до клієнта-власника; зручний перегляд повної історії візитів тварини.

3. Управління записом на прийом

Система має реалізовувати: формування записів на прийом із зазначенням лікаря, дати, часу, типу послуги та додаткових приміток; перевірку зайнятості лікаря та уникнення конфліктів у розкладі; зміну, перенесення та скасування записів; автоматичне інформування клієнтів про створення або зміну запису; перегляд записів у вигляді календаря.

4. Проведення візиту та формування медичної документації

Необхідно забезпечити: створення медичного запису під час візиту; опис симптомів та стану тварини; фіксацію діагнозу, призначень, процедур і рекомендацій; прикріплення результатів аналізів та додаткових файлів; можливість друку або експорту медичних висновків.

5. Управління послугами

Функціональність має включати: створення довідника ветеринарних послуг з описом, ціною та тривалістю; редагування переліку послуг; автоматичний розрахунок вартості візиту за обраними процедурами; формування рахунків клієнтам на основі виконаних послуг.

6. Фінансовий облік

Система повинна підтримувати: створення рахунків за послуги; фіксацію оплат; відстеження заборгованостей; формування фінансових звітів за період.

7. Управління персоналом та ролями

Система має забезпечувати: створення профілів працівників; налаштування ролей (адміністратор, ветеринар, реєстратор); розмежування прав доступу залежно від ролі; ведення робочого розкладу лікарів.

8. Комунікація з клієнтами

Передбачено: автоматичні нагадування про візити (SMS/e-mail); сповіщення про результати аналізів або планові щеплення; можливість ручної розсилки інформаційних повідомлень.

9. Зберігання та захист даних

Система має підтримувати: зберігання даних у структурованій реляційній базі; повнотекстовий пошук; резервне копіювання інформації; аудит дій користувачів та журналювання.

10. Користувацький інтерфейс

До вимог щодо інтерфейсу належать: простота та інтуїтивність взаємодії; адаптивний дизайн для роботи на ПК, планшетах і мобільних пристроях; підтримка української локалізації; розділення інтерфейсів для різних ролей доступу.

Сформульовані функціональні вимоги дозволяють визначити структуру, архітектуру та ключові програмні модулі CRM-системи. Вони забезпечують основу для подальшого проєктування інфологічної моделі, архітектури програмного забезпечення та розроблення інтерфейсу користувача, що розглядається у наступних розділах роботи.

Висновки до першого розділу

Перший розділ роботи був присвячений глибокому аналізу предметної області – діяльності ветеринарного центру, огляду ринку існуючих CRM-рішень, а також чіткій постановці завдання та формуванню детальних функціональних вимог до розроблюваної системи.

Аналіз бізнес-процесів показав, що діяльність ветеринарної клініки є багатокомпонентною і вимагає автоматизації восьми ключових процесів: від реєстрації клієнтів (власників) та обліку пацієнтів (тварин) до планування прийомів, ведення медичної документації (ЕМК), фінансового обліку та комунікації з клієнтами. Виявлено, що неефективність цих процесів через використання паперового або розрізненого електронного обліку призводить до зниження якості обслуговування, втрати даних і ускладнення управління.

Огляд ринку підтвердив, що існуючі рішення поділяються на:

Універсальні CRM (наприклад, Salesforce, Zoho), які є дорогими, складними для невеликого бізнесу та не мають необхідної ветеринарної специфіки (облік тварин, вакцинація).

Спеціалізовані ветеринарні системи (VetCRM, Provect Cloud), які хоча й адаптовані до галузі, часто мають високу вартість, складний інтерфейс і недостатню локалізацію та підтримку української мови.

Медичні інформаційні системи (МІС), які повністю непридатні для ветеринарії через орієнтацію виключно на пацієнтів-людей та державні реєстри.

На основі аналізу були сформульовані детальні функціональні вимоги до системи, згруповані у 10 ключових блоків. Ці вимоги охоплюють усі аспекти діяльності: від створення Електронної Медичної Картки (ЕМК) тварини та управління календарем записів до фінансового обліку, розмежування прав доступу та забезпечення безпеки даних.

2 ПРОЄКТУВАННЯ CRM-СИСТЕМИ ДЛЯ ВЕТЕРИНАРНОГО ЦЕНТРУ

2.1 Інфологічна модель даних та структурні схеми

Проектування системи вимагає розробки концептуальних моделей. Ці моделі слугують «кресленнями» для майбутньої програмної реалізації. Для цього необхідно сформулювати інфологічну модель даних, яка відображає сутність інформації та зв'язки між нею, а також визначити структурну схему, що описує архітектуру та взаємодію основних програмних компонентів.

Інфологічна модель є основним рівнем абстракції в проектуванні баз даних. Її призначення полягає у концептуальному представленні даних, незалежно від конкретної системи управління базами даних (СУБД) чи програмної реалізації. Фактично, ця модель відображає, яка інформація буде зберігатися в системі та як ця інформація пов'язана між собою, використовуючи концепцію сутність-зв'язок (Entity-Relationship, ER-модель).

Сутності (Entities) в інфологічній моделі відповідають ключовим об'єктам, які були ідентифіковані під час аналізу бізнес-процесів ветеринарного центру.

Сутність (Об'єкт)	Опис та ключові атрибути
Клієнт (Власник)	Атрибути: ID (ключ), ПІБ, Телефон, Email, Адреса. <i>Вимога: Зберігання контактної інформації та історії взаємодії.</i>
Пацієнт (Тварина)	Атрибути: ID (ключ), Клієнт_ID (зв'язок), Кличка, Вид, Порода, Дата народження, Стать, Вага, Індивідуальні особливості (алергії). <i>Вимога: Детальний облік тварин та їхній зв'язок із власниками.</i>
Працівник (Лікар)	Атрибути: ID (ключ), ПІБ, Посада, Контактний телефон, Спеціалізація, Роль (для розмежування доступу). <i>Вимога: Управління персоналом та правами.</i>
Прийом (Візит)	Атрибути: ID (ключ), Клієнт_ID (зв'язок), Пацієнт_ID (зв'язок), Лікар_ID (зв'язок), Дата та час, Статус (Заплановано, Виконано, Скасовано), Примітки. <i>Вимога: Основа календарного планування та ведення історії.</i>
Мед. Запис (EHR)	Атрибути: ID (ключ), Прийом_ID (зв'язок), Анамнез, Опис огляду, Діагноз, Призначення, Рекомендації. <i>Вимога: Зберігання структурованої медичної інформації.</i>
Послуга	Атрибути: ID (ключ), Назва послуги, Опис, Ціна, Тривалість. <i>Вимога: Довідник для формування рахунків.</i>
Рахунок	Атрибути: ID (ключ), Прийом_ID (зв'язок), Дата формування, Сума, Статус оплати. <i>Вимога: Фінансовий облік.</i>

Зв'язки описують, як сутності взаємодіють відповідно до бізнес-логіки клініки. Найважливіші зв'язки мають таку кардинальність:

— Клієнт – Пацієнт: один клієнт може мати багато тварин-пацієнтів (один-до-багатьох, 1:M).

- Пацієнт – Прийом: один пацієнт може мати багато візитів/прийомів (один-до-багатьох, 1:M).
- Лікар – Прийом: один лікар проводить багато прийомів (один-до-багатьох, 1:M).
- Прийом – Мед. Запис: кожен прийом має рівно один медичний запис (один-до-одного, 1:1).
- Прийом – Послуга: один прийом може включати багато послуг, і одна послуга може бути надана під час багатьох прийомів (багато-до-багатьох, M:M, реалізується через проміжну таблицю).
- Прийом – Рахунок: кожен прийом генерує рівно один рахунок (один-до-одного, 1:1).

Ці зв'язки забезпечують цілісність даних: якщо ми знаємо ID прийому, ми миттєво можемо знайти лікаря, клієнта, пацієнта, медичний запис і рахунок.

ER-діаграма (Entity-Relationship Diagram), представлена в Додатку А, є графічним відображенням інфологічної моделі даних системи. Вона фіксує всі ключові сутності (об'єкти), необхідні для функціонування ветеринарної клініки, та встановлює логічні зв'язки між ними, відображаючи кардинальність (тип зв'язку).

Ключові сутності та зв'язки:

CLIENT (Клієнт): зберігає дані власника тварини (ПІБ, контакти). Має зв'язок 1:M (один до багатьох) із сутністю PATIENT.

PATIENT (Пацієнт/Тварина): містить детальні дані про тварину (Кличка, Порода, Вага, Алергії). Кожен Пацієнт пов'язаний з одним Клієнтом.

WORKER (Працівник/Лікар): включає інформацію про персонал та його Роль (для розмежування доступу). Зв'язаний із APPOINTMENT як виконавець.

APPOINTMENT (Прийом/Візит): центральна сутність системи, яка пов'язує CLIENT, PATIENT, WORKER та фіксує час/статус події.

MED_RECORD (Медичний Запис): містить результати візиту (Анамнез, Діагноз, Призначення). Має зв'язок 1:1 з APPOINTMENT.

SERVICE (Послуга): довідник усіх послуг із цінами. Зв'язок M:M (багато до

багатьох) з APPOINTMENT реалізований через проміжну таблицю APPOINTMENT_SERVICE.

INVOICE (Рахунок): фінансовий документ, що фіксує суму та статус оплати за конкретний APPOINTMENT.

Ця модель є основою для проєктування реляційної бази даних, забезпечуючи цілісність, несуперечність та ефективність зберігання даних.

Для забезпечення масштабованості, надійності та простоти обслуговування, CRM-систему доцільно проєктувати на основі трирівневої. Ця архітектура чітко розділяє систему на три взаємозалежні рівні, кожен з яких виконує свою унікальну функцію.

Обрана модель розділяє логіку системи таким чином:

Рівень (Tier)	Основна функція	Ключові технології (приклад)
1. Рівень представлення (Presentation Tier)	Забезпечення взаємодії з користувачем (UI/UX).	Веб-браузер (HTML, CSS, JavaScript) або мобільний додаток.
2. Рівень логіки застосунку (Application Tier)	Обробка бізнес-логіки, виконання вимог CRM.	Сервер застосунків (наприклад, Node.js, Python/Django, .NET) та API.
3. Рівень даних (Data Tier)	Зберігання та управління інформацією.	Реляційна СУБД (наприклад, PostgreSQL, MySQL) та інфологічна модель.

Структурна схема системи відображає потік даних та ролі компонентів у процесі обробки запитів:

Клієнтський інтерфейс (Presentation Tier).

Роль: відображає дані та дозволяє користувачам (адміністраторам, лікарям) вводити інформацію (наприклад, запис на прийом, опис візиту).

Взаємодія: надсилає запити (наприклад, "Створити новий прийом") через інтернет до Рівня логіки застосунку (API).

Сервер застосунків (Application Tier).

Роль: це «мозок» системи. Він приймає запити від клієнтського інтерфейсу, перевіряє права доступу користувача, виконує бізнес-логіку (наприклад, перевіряє конфлікти у розкладі, розраховує вартість послуг) і формує запит до бази даних.

Взаємодія: обробляє запити, викликає функції обробки даних (CRUD-операції) на Рівні даних і повертає відформатовану відповідь до клієнтського інтерфейсу.

Сервер баз даних (Data Tier).

Роль: надійне зберігання всіх даних відповідно до розробленої інфологічної моделі.

Взаємодія: приймає запити лише від Сервера застосунків, виконує їх і повертає необхідні дані. Прямий доступ з клієнтського інтерфейсу до цього рівня заборонено з міркувань безпеки та цілісності даних.

Така архітектура гарантує, що система буде гнучкою для майбутніх змін у бізнес-логіці та надійною для захисту конфіденційної інформації клієнтів і медичних записів.

Додаток В містить структурні та функціональні діаграми, що описують архітектуру та взаємодію компонентів системи. Зокрема на рис. В. 1. представлено контекстну діаграму CRM-системи ветеринарного центру (DFD Level 0), яка відображає систему як єдиний чорний ящик і показує її взаємодію із зовнішніми Акторами (Клієнт, Лікар, Адміністратор), а також основні потоки даних, що перетинають межі системи. Клієнт надсилає запити на запис, оновлення даних та отримує нагадування. Лікар/Працівник отримує графік прийомів та історію пацієнта, а надає системі медичні записи та результати огляду. Адміністратор керує користувачами, послугами, та отримує статистику і фінансові звіти.

На рис. В. 2. Додат. В. представлена деталізація внутрішніх процесів CRM-системи ветеринарного центру – ця діаграма (DFD) деталізує внутрішні процеси системи та їхню взаємодію з Базою даних. Вона демонструє, як ключові процеси (позначені P1–P7) обробляють інформацію:

P1. Управління контактами та P2. Управління пацієнтами відповідають за збір та збереження даних Клієнтів та Пацієнтів у Базі даних.

P3. Запис на прийом та P7. Повідомлення та нагадування керують календарем та комунікацією.

P4. Проведення візиту і медичний запис та P5. Формування рахунку є ядром клінічних та фінансових процесів, що генерують Медичний запис та Рахунок.

Діаграма підтверджує, що База даних є єдиним централізованим сховищем інформації для всіх функціональних модулів системи.

На рис. В. 3. Додатка В представлена діаграма варіантів використання CRM-системи ветеринарного центру, яка показує функціональні можливості системи з точки зору Акторів. Вона ілюструє, які саме дії (варіанти використання) може виконувати кожен користувач (Ветеринар, Клієнт/Власник, Адміністратор). Адміністратор має найширший функціонал: керування базою, розкладом, фінансами (Сформувати Рахунок) та довідником Послуг. Ветеринар орієнтований на медичну частину: Створити ЕМК (Медичний Запис), Переглянути історію візитів. Клієнт взаємодіє із зовнішніми точками: Записати на прийом. Ця діаграма є важливим інструментом для візуалізації функціональних вимог та визначення меж доступу для різних ролей користувачів.

Отже, на етапі проєктування було успішно розроблено концептуальні моделі, які слугують логічними «кресленнями» для подальшої програмної реалізації CRM-системи.

Сформована інфологічна модель даних (ER-модель) чітко визначила сім ключових сутностей (Клієнт, Пацієнт, Працівник, Прийом, Мед. Запис, Послуга, Рахунок). Були встановлені необхідні логічні зв'язки між цими сутностями з відповідною кардинальністю (1:M, 1:1, M:M), що гарантує цілісність, несуперечність та ефективну структуру даних у майбутній реляційній базі.

ER-діаграма (Додаток А) є прямою і наочною основою для створення фізичної моделі бази даних. Для забезпечення надійності, гнучкості та масштабованості системи обрано Трирівневу архітектуру (рівень представлення, рівень логіки застосунку та рівень даних). Таке розділення чітко ізолює бізнес-логіку (Сервер застосунків) від користувацького інтерфейсу та сховища даних (Сервер баз даних). Це підвищує безпеку, спрощує обслуговування та гарантує гнучкість системи до майбутніх змін. Розроблені діаграми DFD (контекстна та деталізована, Додаток В) візуалізують потоки даних та внутрішні процеси CRM-системи.

Діаграма варіантів використання (UML Use Case) (Додаток В, рис. В.3) визначила функціональні межі системи та підтвердила правильність розмежування ролей між основними Акторами (Адміністратор, Ветеринар,

Клієнт/Власник).

Таким чином, розроблені інфологічна модель та структурна схема повністю задовольняють вимоги до концептуального проєктування, забезпечуючи надійний та логічно обґрунтований фундамент для переходу до етапу технічного проєктування та програмної реалізації системи.

2.2 Проєктування архітектури, REST-інтерфейсів та функціональних модулів

Цей етап включає визначення архітектурного шаблону, проєктування механізму взаємодії між компонентами за допомогою RESTful API та структурування системи на функціональні модулі. На основі вимог до масштабованості, надійності та простоти супроводу системи, у попередньому підрозділі обґрунтовано вибір класичного трирівневого архітектурного шаблону. Цей підхід забезпечує ефективну декомпозицію системи, мінімізує зв'язність між компонентами та дозволяє незалежно розвивати та оптимізувати кожен рівень.

Рівень даних – цей рівень відповідає за перманентне зберігання даних та забезпечення реляційної цілісності інформації. Його архітектура базується на інфологічній моделі, представлений у Додатку А. Вибір СУБД: З огляду на необхідність зберігання структурованих та взаємопов'язаних даних (Клієнт-Пацієнт-Прийом-Медичний Запис), обрано реляційну СУБД (PostgreSQL або MySQL). Це гарантує ефективну обробку складних запитів та підтримку транзакцій, що критично для фінансового та медичного обліку.

Рівень логіки застосунку – є сервісно-орієнтованим шаром, що реалізує всю бізнес-логіку системи. Його головна функція – обробка запитів від клієнтського рівня, валідація даних, виконання обчислень (наприклад, формування вартості послуг) та управління транзакціями до Баз даних.

Інтерфейс взаємодії: взаємодія з цим рівнем здійснюється виключно через RESTful API (Representational State Transfer Application Programming Interface), що дозволяє використовувати його як єдине джерело даних для різних клієнтів (веб, мобільний додаток).

Технологія: будемо для розробки використовувати Python та Node.js, які сприяють швидкій та надійній розробці API.

Рівень представлення – забезпечує графічний інтерфейс користувача (GUI) та керує взаємодією з Акторами (Адміністратором, Ветеринаром).

Технологія: Використовується парадигма односторінкового застосунку (SPA) на базі сучасних JavaScript-бібліотек, таких як React або Vue.js, що забезпечує високу швидкість відгуку та адаптивний дизайн (рис. 3.1).

Проектування REST-інтерфейсів є критично важливим для забезпечення слабкої зв'язності між рівнями. API реалізує принцип CRUD (Create, Read, Update, Delete) для кожної основної сутності системи.

Таблиця 2.1. Приклади основних REST-ресурсів

Ресурс (URI)	HTTP-метод	Опис операції (Бізнес-логіка)
/api/clients	POST	Створення нового профілю клієнта (ПІБ, контакти).
/api/clients/{id}	GET	Отримання детальних даних клієнта, включаючи список пов'язаних Пацієнтів.
/api/patients	POST	Реєстрація нової тварини з прив'язкою до існуючого клієнта (Власника).
/api/appointments	POST	Запис на прийом: валідація часу та зайнятості лікаря (Р3. Запис на прийом).
/api/appointments/{id}/record	POST	Створення Медичного Запису (ЕМК) після візиту (Р4. Проведення візиту).
/api/invoices	POST	Генерація рахунку на основі послуг, наданих під час Прийому (Р5. Формування рахунку).

Використання REST-архітектури забезпечує чіткість, стандартизацію та незалежність рівнів, дозволяючи оновлювати Frontend без необхідності зміни Backend і навпаки.

Розроблена CRM-система логічно поділяється на функціональні модулі, які відповідають ключовим бізнес-процесам (DFD) та пунктам головного меню (див. табл. 2.2.).

Таблиця 2.2. Структура функціональних модулів

№	Функціональний модуль	Опис та зв'язок з бізнес-процесом
1.	Облік клієнтів та пацієнтів	P1, P2. Централізоване керування базою власників та їхніх тварин. Дозволяє швидко вводити, редагувати та шукати профілі.

2.	Календар та прийоми	P3. Модуль для управління розкладом лікарів, запису на прийом, перенесення та скасування візитів. Тут відбувається первинна перевірка конфліктів.
3.	Медичні картки (емк)	P4. Ведення електронної історії хвороби. Функціонал для фіксації анамнезу, діагнозу, призначень та прикріплення результатів досліджень.
4.	Фінанси та облік послуг	P5, P6. Створення довідника послуг, автоматичне формування рахунків на основі ЕМК та облік оплат.
5.	Комунікації та Сповідання	P7. Модуль для генерації та відправки автоматичних SMS/Email нагадувань про майбутні візити та вакцинацію.
6.	Персонал та адміністрування	Внутрішній модуль для управління обліковими записами, ролями (Ветеринар, Адміністратор) та розмежуванням доступу.

Така модульна структура забезпечує логічну ізоляцію функцій, що спрощує розробку, тестування та подальше розширення системи.

Функціональні модулі системи безпосередньо відповідають ключовим варіантам використання (Додаток В) та забезпечують повну автоматизацію бізнес-процесів ветеринарної клініки.

Модуль «Облік Клієнтів та Пацієнтів». Призначення: централізоване управління профілями власників та їхніх тварин (процеси P1 та P2). Функціональність: створення, редагування та пошук даних. Для кожного Пацієнта (Тварини) забезпечується фіксація детальних параметрів (вид, порода, дата народження, вага) та критичної медичної інформації (алергії, історія вакцинацій).

Модуль «Календар та Управління Прийомами». Призначення: ефективне планування робочого часу персоналу та координація візитів. Функціональність: візуалізація розкладу (день/тиждень/місяць), створення нових записів, динамічна зміна статусу (наприклад, "Очікує", "На прийомі", "Виконано"). Модуль інтегрований з API для перевірки зайнятості Лікаря.

Модуль «Електронна Медична Картка (ЕМК)». Призначення: ведення структурованої медичної документації (процес P4). Функціональність: дозволяє Лікарю створювати детальний Медичний Запис після прийому: фіксація анамнезу, результатів огляду, діагнозу, призначень та рекомендацій. Забезпечується можливість прикріплення зовнішніх файлів (рентгенівські знімки, результати аналізів).

Модуль «Фінанси та Облік Послуг». Призначення: автоматизація фінансових операцій (процеси P5 та P6). Функціональність: управління

довідником послуг та їхніми тарифами. Автоматичне формування рахунків на основі даних про виконані послуги під час візиту, фіксація оплат та контроль заборгованостей.

Модуль «Адміністрування та Звіти». Призначення: внутрішній контроль та управління системою (функції Адміністратора). Функціональність: налаштування профілів працівників, присвоєння ролей та розмежування прав доступу. Генерація статистичних та фінансових звітів (наприклад, дохід за період, навантаження лікарів) для управлінського аналізу.

Сформована структура забезпечує повноцінну підтримку всіх бізнес-процесів ветеринарного центру та створює надійну технічну базу для подальшої програмної реалізації.

2.3 Реалізація основних модулів системи та їх взаємодія

Процес реалізації основних модулів системи передбачає перетворення детально розроблених технічних специфікацій та функціональний програмний код. Він охоплює кодування функціональних модулів, забезпечення їх взаємодії через RESTful API та інтеграцію з рівнем даних.

Реалізація системи виконується відповідно до трирівневої архітектури, що забезпечує чітке розділення відповідальності між компонентами. Взаємодія між модулями відбувається за принципом «клієнт-сервер», де Рівень логіки застосунку (Backend) виступає як центральний хаб, що обробляє всі запити, які надходять від Рівня представлення (Frontend).

Таблиця 2.3. Технологічний стек та функції архітектурних рівнів

Рівень	Основна функція	Технологічна платформа (Приклад)	Механізм взаємодії
Рівень представлення (Frontend)	Забезпечення графічного інтерфейсу (UI/UX) та ініціація запитів.	React/Vue.js	Асинхронні HTTP-запити
Рівень логіки застосунку (Backend)	Обробка бізнес-логіки, валідація, авторизація, управління транзакціями.	Python/Django REST Framework	RESTful API

Рівень даних (Database)	Сховище структурованих даних, виконання CRUD- операцій.	PostgreSQL/MySQL	SQL-запити
----------------------------	---	------------------	------------

Взаємодія між рівнями реалізується за принципом «тонкий клієнт – товстий сервер», де клієнтський інтерфейс лише надсилає HTTP-запити до API-сервера, який містить усю ключову бізнес-логіку.

Розглянемо як приклад, принцип реалізації транзакційної взаємодії. Зокрема, критичні операції (наприклад, створення рахунку або внесення медичного запису) реалізуються як атомарні транзакції на рівні СУБД. Наприклад, процес формування медичного запису та одночасне оновлення статусу прийому вимагають успішного виконання обох операцій, або ж відкату всіх змін (ROLLBACK) у разі помилки.

Приклад: Функція Django (Backend) для створення Медичного Запису та оновлення статусу Прийому

```
from django.db import transaction

@transaction.atomic
def create_medical_record_and_update_appointment(appointment_id, data):
    # 1. Створення Медичного Запису (Med_Record)
    medical_record = MedRecord.objects.create(appointment_id=appointment_id,
**data)

    # 2. Оновлення статусу Прийому
    appointment = Appointment.objects.get(pk=appointment_id)
    appointment.status = 'Виконано'
    appointment.save()

    return medical_record
```

Реалізація кожного модуля забезпечує відповідність логіці, визначеній у функціональних вимогах та інфологічній моделі (Додаток А).

Модуль «Облік Клієнтів та Пацієнтів» (P1, P2)

Механізм: реалізується як два пов'язаних REST-ресурси (/clients та /patients).

Ключова логіка: при реєстрації нового пацієнта (тварини) система на рівні Backend (модель/валідатор) примусово перевіряє наявність та валідність зовнішнього ключа (Клієнт_ID), забезпечуючи реляційну цілісність зв'язку 1:М.

Приклад: JavaScript-код (Frontend) для відправки запиту на реєстрацію Пацієнта

```
const patientData = {
  owner_id: 123, // ID клієнта
  name: 'Рекс',
  species: 'Собака',
  breed: 'Лабрадор',
  allergies: 'Злаки'
};

axios.post('/api/patients', patientData)
  .then(response => console.log("Пацієнт успішно зареєстрований"))
  .catch(error => console.error("Помилка валідації:", error.response.data));
```

Модуль «Календар та Прийоми» (P3)

Механізм: використовує ресурс /appointments.

Ключова логіка: основна складність полягає у валідації розкладу. При спробі створення нового прийому, Backend виконує запит до БД для перевірки, чи перетинається (overlap) новий часовий інтервал із вже існуючими записами для обраного Лікар_ID. У разі конфлікту, API повертає помилку 409 (Conflict).

Модуль «Електронна Медична Картка (ЕМК)» (P4)

Механізм: використовує ресурс /appointments/{id}/record для ініціації запису в таблицю MED_RECORD.

Ключова логіка: внесення медичних даних дозволено лише користувачам із роллю "Ветеринар" (перевірка ролі виконується за JWT-токеном) та лише для

прийомів, що мають статус "На прийомі". Це забезпечує валідність робочого процесу.

Модуль «Фінанси та Облік Послуг» (P5, P6)

Механізм: взаємодіє з ресурсами /services та /invoices.

Ключова логіка: після завершення медичного обслуговування, система автоматично генерує об'єкт Рахунок (INVOICE). Розрахунок суми відбувається шляхом агрегації цін усіх пов'язаних послуг з довідника SERVICE, забезпечуючи фінансову точність.

Модуль «Комунікації та Сповіщення» (P7)

Механізм: реалізується за допомогою асинхронної обробки та інтеграції із зовнішніми API.

Ключова логіка: замість синхронного блокування основного сервера, використовується брокер повідомлень (наприклад, Redis або RabbitMQ) та фоновий процес (Worker). Цей процес регулярно виконує запит до БД на виявлення необхідності відправлення нагадувань (за 24 години до прийому) та ініціює виклики до API зовнішнього SMS-шлюзу.

Надійність системи забезпечується двома ключовими механізмами:

Автентифікація та Авторизація: використовується стандарт JSON Web Tokens (JWT). Після успішного входу користувач отримує токен, який містить інформацію про його Роль (Ветеринар, Адміністратор). Кожен наступний REST-запит перевіряється на рівні Middleware (проміжне програмне забезпечення) на наявність та валідність токена.

Розмежування прав доступу: на основі Ролі, отриманої з JWT, реалізуються контролери, які обмежують доступ до певних операцій. Наприклад, лише користувачі з роллю "Адміністратор" можуть виконувати DELETE-операції над сутностями CLIENT або SERVICE.

Таким чином, реалізація системи на базі архітектурного паттерну «RESTful API – тонкий клієнт» та модульної структури забезпечує високу продуктивність, надійність транзакцій та чітке розмежування відповідальності між компонентами. Такий підхід створює ефективну та гнучку платформу для експлуатації CRM-

системи у ветеринарному центрі.

Висновки до другого розділу

У другому розділі виконано повний цикл проектування CRM-системи, що трансформувало функціональні вимоги та результати аналізу предметної області в детальні технічні специфікації, придатні для безпосередньої програмної реалізації.

Розроблено інфологічну модель даних у форматі ER-діаграми (Додаток А, Рис. А.1), яка ідентифікувала сім ключових сутностей (Клієнт, Пацієнт, Прийом, Мед. Запис, Рахунок та ін.) та визначила їхні логічні зв'язки (1:М, М:М, 1:1). Це забезпечує реляційну цілісність, несуперечність та ефективність структури майбутньої бази даних.

Обґрунтовано вибір трирівневого архітектурного шаблону (Рівні Представлення, Логіки Застосунку та Даних). Цей поділ гарантує слабку зв'язаність, масштабованість та незалежність компонентів.

Розроблено DFD-діаграми (Рівні 0 та 1, Додаток В, Рис. В.1, В.2) та UML Діаграму варіантів використання (Додаток В, Рис. В.3), які чітко візуалізували потоки даних та розмежування функціональних обов'язків між Акторами (Лікар, Адміністратор, Клієнт).

Деталізовано взаємодію рівнів за допомогою RESTful API, визначивши стандартизовані ресурси та HTTP-методи для операцій CRUD над основними сутностями (Таблиця 2.1).

Систему структуровано на шість функціональних модулів (Таблиця 2.2), кожен з яких відповідає за автоматизацію ключового бізнес-процесу ветеринарного центру (наприклад, «Календар та Прийоми», «Електронна Медична Картка»).

Обґрунтовано використання PostgreSQL/MySQL для рівня даних та сучасних фреймворків (Python/Django, React) для логіки та представлення, а також ключові механізми безпеки (JWT, RBAC) та транзакційної цілісності.

3 РОЗРОБЛЕННЯ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ ТА ОЦІНЮВАННЯ РОБОТИ СИСТЕМИ

3.1 Опис інтерфейсу CRM-системи та сценарії взаємодії користувача

Інтерфейс системи реалізований за принципом односторінкового застосунку (SPA) з використанням сучасних JavaScript-бібліотек (React). Це забезпечує швидке завантаження та динамічний відгук без перезавантаження сторінок.

Основні елементи інтерфейсу:

Головна Навігаційна Панель (Sidebar): вертикальна панель ліворуч, що містить основні функціональні модулі відповідно до ролі користувача (див. Таблицю 2.2).

Панель Керування (Dashboard): домашня сторінка системи, що містить агреговану інформацію, ключові метрики (KPI) та швидкий доступ до поточних завдань (наприклад, список очікуваних прийомів).

Робоча Область (Content Area): центральний простір, де відображаються форми, таблиці даних, календар та медичні картки.

Управління обліковими записами персоналу (Ветеринари, Адміністратори) реалізується централізовано на рівні Модуля Адміністрування. Для забезпечення безпеки та атомарності операцій, процедура створення облікових записів ініціюється виключно Адміністратором системи.

Обрана політика управління доступом є оптимальною для бізнес-моделі ветеринарної клініки та базується на принципах ізоляції даних та мінімально необхідного доступу (Principle of Least Privilege):

Контрольоване ведення документації: ключовим завданням системи є ведення Електронної Медичної Картки (ЕМК). Персонал клініки веде всі записи, медичну документацію та фінансовий облік. Це забезпечує високий рівень контролю над якістю та цілісністю введених даних.

Зниження ризиків безпеки: обмеження доступу Клієнтів до внутрішнього інтерфейсу системи мінімізує вектори атак та потенційні ризики несанкціонованого доступу до конфіденційної інформації інших клієнтів.

У відповідності до підвищених вимог до ергономіки та прозорості взаємодії,

функціонал системи було розширено реалізацією *Порталу самообслуговування Клієнта*. Цей модуль забезпечує Клієнтам можливість автономного доступу до персоналізованої інформації про своїх тварин та історію обслуговування.

Процес створення облікових записів Клієнтів реалізується через розширений інтерфейс реєстрації, що знаходиться під контролем персоналу:

Інтерфейс реєстрації: до форми створення Клієнта додано опцію "Створити обліковий запис для доступу до особистого кабінету".

- Автоматизоване провізйонування: активація цієї опції ініціює виклик Edge Function (Serverless Function) на рівні Backend для безпечної та атомарної генерації облікового запису (email та хешованого пароля).

- Процедура активації: адміністратор/реєстратор фіксує email та пароль і передає ці дані Клієнту, завершуючи процес надання доступу.

Функціональність Порталу Клієнта (ClientPortal).

ClientPortal функціонує як окремий веб-застосунок (Frontend), доступний Клієнтам після успішної автентифікації. Він реалізує наступний функціонал у режимі тільки для читання (Read-Only):

- Інформаційна панель: надає агреговану статистику, включаючи кількість зареєстрованих тварин, загальну кількість прийомів та список відкритих рахунків.

- Облік пацієнтів: детальний перелік усіх тварин, прив'язаних до облікового запису, з можливістю перегляду повної історії візитів та медичних записів (MED_RECORD).

- Управління прийомами: повна історія запланованих та виконаних візитів зі статусами.

- Фінансовий огляд: список виставлених рахунків (INVOICE) з відображенням суми та статусу оплати.

Динамічний розподіл доступу

Система використовує механізм динамічного розподілу доступу на основі ролей:

- Персонал: користувачі з ролями «Адміністратор» або «Ветеринар» отримують доступ до повного інтерфейсу управління клінікою (CRM-системи) та

мають права READ/WRITE/DELETE.

- Клієнт: користувачі з роллю «Клієнт» автоматично перенаправляються до ClientPortal, де їхній доступ суворо обмежений режимом READ-ONLY на основі впроваджених RLS-політик.

Таким чином, розширення функціоналу за рахунок ClientPortal підвищує прозорість обслуговування, не компрометуючи при цьому безпеку та цілісність внутрішньої CRM-системи. А реалізований механізм управління користувачами забезпечує чітке застосування RBAC-моделі та відповідає високим вимогам до безпеки та оперативної ефективності ветеринарного центру.

Сценарій взаємодії для ролі «Адміністратор»

Адміністратор є ключовим користувачем для управління клієнтською базою, плануванням та фінансами. Його інтерфейс орієнтований на швидкість введення та обробки даних.

Сценарій 1: реєстрація нового Клієнта та Пацієнта

Мета: швидко внести дані нового власника та його тварини.

Дія 1: адміністратор обирає "Облік Клієнтів" → "Додати Клієнта".

Дія 2: заповнюється форма реєстрації Клієнта (ПІБ, телефон, email) (Рис. 3.2).

Дія 3: після успішного збереження, система автоматично пропонує форму для додавання Пацієнта, пов'язуючи його з щойно створеним Клієнтом (Рис. 3.3).

Системний результат: Frontend надсилає послідовно два POST-запити до /api/clients та /api/patients, забезпечуючи реляційну цілісність (Клієнт_ID є зовнішнім ключем).

Сценарій 2: запис на прийом та Валідація розкладу

Мета: запланувати прийом із перевіркою зайнятості лікаря.

Дія 1: адміністратор обирає "Календар" і клікає на бажаний часовий слот.

Дія 2: у формі запису обирається Клієнт, Пацієнт та Лікар.

Дія 3: при збереженні система виконує валідацію розкладу (Backend-логіка РЗ).

Системний результат: якщо у Лікаря є інший прийом, що перетинається

(overlap) у часі, Backend повертає HTTP-помилку 409 Conflict. Інтерфейс відображає користувачеві відповідне сповіщення, пропонуючи вибрати інший час або лікаря.

Сценарії взаємодії для ролі «Ветеринар»

Інтерфейс ветеринара зосереджений на медичній документації, швидкому доступі до історії хвороби та формуванні ЕМК.

Сценарій 3: доступ до Медичної Картки (ЕМК)

Мета: швидко отримати повну історію пацієнта перед початком прийому.

Дія 1: ветеринар переходить на Панель Керування (Dashboard) і бачить список поточних прийомів.

Дія 2: лікар обирає Пацієнта зі списку (статус "На прийомі").

Дія 3: Система відкриває Карту Пацієнта, де за допомогою вкладок відображаються:

Основні дані Пацієнта (порода, вага, алергії).

Історія всіх минулих Візитів та Медичних Записів.

Системний результат: Frontend виконує GET-запит до `/api/patients/{id}` та вкладений запит до `/api/appointments?Patient_id={id}` для отримання повної історії.

Сценарій 4: створення Медичного Запису та Формування Рахунку

Мета: документувати візит, призначити послуги та підготувати фінансовий документ.

Дія 1: під час прийому Ветеринар відкриває відповідний Прийом і переходить у режим "Створити ЕМК" (P4).

Дія 2: заповнюються структуровані поля: Анамнез, Результати огляду, Діагноз.

Дія 3: у розділі "Призначені Послуги" Ветеринар обирає послуги з довідника (наприклад, "Консультація", "Вакцинація").

Дія 4: після завершення і натискання кнопки "Завершити візит", система виконує:

Створення запису у MED_RECORD.

Оновлення статусу APPOINTMENT на "Виконано".

Ініціювання автоматичної генерації Рахунку (INVOICE) на рівні Backend (P5), що агрегує вартість обраних послуг.

Ці сценарії відображають, як UI/UX CRM-системи забезпечує ефективну та логічно послідовну підтримку основних бізнес-процесів ветеринарного центру.

Сценарії взаємодії для ролі «Клієнт (Власник тварини)»

Взаємодія Клієнта з CRM-системою відбувається здебільшого через два канали: клієнтський веб-портал (для активних дій) та автоматизовані сповіщення (для пасивних дій). Ці сценарії відповідають вимогам комунікації та самообслуговування (Пункт P3, P7 з DFD, Додаток В).

Сценарій 5: Самостійний онлайн-запис на прийом

Мета: клієнт самостійно бронює час для візиту без участі Адміністратора.

Дія 1: клієнт відкриває зовнішню веб-форму (Портал Клієнта) та автентифікується.

Дія 2: клієнт обирає зі списку своїх тварин-пацієнтів та бажаного Лікаря.

Дія 3: система відображає доступні часові слоти Лікаря, виключивши вже заброньовані.

Дія 4: клієнт підтверджує запис.

Системний результат: Frontend надсилає POST-запит до /api/appointments. Backend виконує валідацію, реєструє прийом та, якщо запис успішний, ініціює миттєве сповіщення Клієнту.

Сценарій 6: Отримання та підтвердження Нагадування

Мета: Система автоматично нагадує Клієнту про майбутній візит.

Дія 1: за 24 години до запланованого Прийому (згідно з логікою P7), фоновий процес на Рівні логіки застосунку ініціює відправку нагадування.

Дія 2: клієнт отримує SMS або email-сповіщення, що містить деталі візиту та посилання для підтвердження або скасування.

Дія 3 (Опціонально): клієнт переходить за посиланням, щоб підтвердити свою присутність.

Системний результат: якщо Клієнт підтверджує, система через API виконує

PATCH-запит до `/api/appointments/{id}` для оновлення поля Статус на "Підтверджено". Це зменшує ризик неявки.

Сценарій 7: Перегляд історії візитів та медичних даних

Мета: клієнт отримує доступ до медичної інформації своїх тварин.

Дія 1: клієнт авторизується на Порталі та обирає розділ "Мої Пацієнти".

Дія 2: клієнт обирає конкретну тварину та переглядає список усіх минулих Прийомів.

Дія 3: для кожного завершеного Прийому Клієнт може переглянути відповідний Медичний Запис (Діагноз, Призначення) та завантажити пов'язаний Рахунок (INVOICE).

Системний результат: Frontend виконує GET-запити до `/api/patients/{id}/records` та `/api/appointments/{id}/invoice`, фільтруючи доступ до даних, щоб Клієнт бачив лише власну інформацію.

Впровадження цих сценаріїв самообслуговування значно знижує навантаження на Адміністратора та підвищує загальну задоволеність Клієнтів якістю обслуговування.

3.2 Розгортання, налаштування, тестування та керівництво користувача

Фінальний етап проєктування та впровадження CRM-системи охоплює переведення розробленого програмного забезпечення з середовища розробки у робоче середовище, забезпечення його якості та створення супровідної документації для кінцевих користувачів.

Розгортання системи виконується згідно з обраною трирівневою архітектурою та передбачає розподіл компонентів між окремими серверами чи контейнерами для забезпечення ізоляції та масштабованості.

Для розгортання використовується стратегія контейнеризації за допомогою технології Docker. Це дозволяє уніфікувати середовища розробки, тестування та продуктивного розгортання, мінімізуючи проблеми, пов'язані з несумісністю операційних систем або залежностей.

– Контейнери: створюються окремі контейнери для:

- Рівня даних: СУБД PostgreSQL (або MySQL).
- Рівня логіки застосунку: Backend-сервер (Django/Node.js).
- Рівня представлення: Frontend-застосунок (React/Vue.js), який обслуговується веб-сервером (наприклад, Nginx).

– Оркестрація: для управління взаємодією та залежностями між контейнерами використовується Docker Compose (для невеликих систем) або Kubernetes (для складніших та масштабованих середовищ).

– Продуктивне середовище: система розгортається на хмарному хостингу (наприклад, AWS, Google Cloud або VDS), що забезпечує високу доступність та можливість горизонтального масштабування.

Після успішного розгортання контейнерів виконуються первинні налаштування:

– Налаштування СУБД: створення бази даних та користувача з необхідними правами доступу.

– Міграція схеми: запуск міграційних скриптів Backend для створення усіх таблиць (CLIENT, PATIENT, APPOINTMENT, MED_RECORD тощо) відповідно до ER-діаграми (Додаток А).

– Ініціалізація даних: внесення початкових даних:

– Створення облікового запису адміністратора (суперкористувача).

– Завантаження довідників (наприклад, перелік SERVICE та WORKER).

– Конфігурація API: встановлення змінних середовища для підключення до зовнішніх сервісів (SMS-шлюз, SMTP-сервер).

Тестування є обов'язковим та інтегральним етапом життєвого циклу розробки програмного забезпечення, що забезпечує верифікацію (відповідність розробленого продукту специфікаціям) та валідацію (відповідність продукту очікуванням користувача). Результатом цього процесу є підтвердження відповідності CRM-системи як функціональним, так і нефункціональним вимогам, визначеним на попередніх етапах проєктування.

Для забезпечення комплексної перевірки системи застосовується багаторівнева стратегія тестування, яка включає низку послідовних технік:

Модульне тестування (Unit Testing): цей рівень фокусується на перевірці ізольованих функцій та методів на Рівні логіки застосунку (Backend). Наприклад, обов'язковому тестуванню підлягають функції, відповідальні за коректний розрахунок вартості послуг чи алгоритми валідації розкладу прийомів.

Інтеграційне тестування (Integration Testing): метою інтеграційного тестування є перевірка коректної взаємодії між різними модулями та архітектурними рівнями системи. Зокрема, проводиться тестування маршрутів RESTful API; наприклад, POST-запит до ресурсу `/api/appointments` перевіряє не лише факт створення запису, а й коректну обробку конфлікту розкладу.

Системне тестування (System Testing): на цьому етапі виконується комплексна перевірка функціональності системи в цілому. Тестування проводиться відповідно до сценаріїв використання, визначених для кожного Актора (Адміністратор, Ветеринар, Клієнт), що підтверджує, що система функціонує як єдиний, узгоджений механізм.

Для ілюстрації процесу валідації критичних бізнес-правил застосовані наступні тестові сценарії (Таблиця 3.1):

Таблиця 3.1. Тестові сценарії

Сценарій тестування	Мета перевірки	Очікуваний результат реакції системи
T1. Тест реляційної цілісності	Перевірка неможливості створити сутність <code>PATIENT</code> (Пацієнт) без посилання на існуючий <code>CLIENT_ID</code> (Клієнт).	Система повертає помилку <code>400 Bad Request</code> або <code>500 Internal Server Error</code> , інформуючи про порушення обмеження зовнішнього ключа (<code>ForeignKey Constraint Failed</code>).
T2. Тест конфлікту прийому	Моделювання спроби записати Лікаря на час, який уже зайнятий, відповідно до логіки процесу P3 .	Система повертає стандартизований HTTP-статус 409 Conflict (згідно з п. 3.1.2, Сценарій 2).
T3. Тест ролі (RBAC)	Спроба виконання DELETE -операції над сутністю <code>SERVICE</code> користувачем з роллю «Ветеринар».	Система повертає HTTP-статус 403 Forbidden , що підтверджує успішне застосування механізму розмежування доступу.
T4. Тест фінансової транзакції	Перевірка коректності автоматичної генерації сутності <code>INVOICE</code> (Рахунок)	Успішне створення запису в таблиці <code>INVOICE</code> з точною агрегованою сумою наданих послуг.

	після завершення APPOINTMENT (логіка P5).	
--	---	--

Завершальним етапом впровадження є забезпечення ергономічності та ефективності експлуатації системи, що досягається розробкою супровідної документації та проведенням навчання.

З метою забезпечення самостійної роботи персоналу було розроблено керівництво користувача, яке є детальною інструкцією, що описує функціональні можливості системи та послідовність дій для виконання типових завдань (додаток Г).

Таким чином, успішне виконання етапів розгортання, тестування та забезпечення супровідної документації гарантує готовність CRM-системи до продуктивної експлуатації та ефективного використання у ветеринарному центрі.

Висновки до третього розділу

У третьому розділі було завершено проєктування та деталізацію рівня представлення CRM-системи, а також визначено ключові аспекти її впровадження, тестування та супроводу.

Інтерфейс системи реалізовано на базі архітектурного підходу Single Page Application (SPA) із застосуванням сучасних бібліотек (React), що забезпечує високу динамічність та швидкість відгуку. Чітко визначено структуру інтерфейсу (Sidebar, Dashboard, Робоча Область) та розроблені детальні сценарії взаємодії (Use Cases) для ключових ролей: Адміністратор (реєстрація, планування), Ветеринар (ведення ЕМК, формування рахунків) та Клієнт (самообслуговування). Це гарантує ефективну підтримку всіх бізнес-процесів клініки.

Впроваджено сувору політику Розмежування прав доступу на основі ролей (RBAC) для персоналу, що базується на принципі мінімально необхідного доступу. Функціонал системи було розширено за рахунок реалізації Порталу самообслуговування Клієнта (ClientPortal). Цей окремий модуль дозволяє Клієнтам автономно переглядати власні дані (історію прийомів, медичні записи, рахунки) у режимі Read-Only.

Безпека даних Клієнтів у ClientPortal забезпечена завдяки впровадженню політик Row-Level Security (RLS) та використання Edge Functions для атомарного провізюнування облікових записів. Це забезпечує прозорість обслуговування без компрометації внутрішньої цілісності CRM.

Для розгортання обрано стратегію контейнеризації (Docker), що гарантує ізоляцію компонентів (СУБД, Backend, Frontend) та полегшує масштабування системи у хмарному середовищі. Розроблена багаторівнева стратегія тестування (Модульне, Інтеграційне, Системне) та визначені тестові сценарії (T1-T4) для верифікації критичних бізнес-правил (наприклад, конфлікт розкладу, цілісність даних, фінансова транзакція).

Створено основу керівництва користувача та визначено програму практичного навчання персоналу, що є необхідною умовою для успішної продуктивної експлуатації системи.

ВИСНОВКИ

У рамках виконання дипломної роботи було досягнуто головної мети – розроблено архітектуру, інформаційну модель та функціональну реалізацію CRM-системи для автоматизації бізнес-процесів ветеринарного центру. Проведене дослідження комплексно охопило етапи аналізу предметної області, проєктування логічної структури даних, розроблення архітектури системи, реалізації програмних модулів і тестування роботи програмного забезпечення. Отримані результати підтверджують наукову обґрунтованість і практичну цінність запропонованого рішення.

Аналітичний етап забезпечив формування концептуального підґрунтя для побудови CRM-системи. Так, під час роботи над дослідженням було:

- виконано аналіз бізнес-процесів ветеринарного центру та визначено ключові інформаційні потреби, пов'язані з реєстрацією клієнтів, обліком пацієнтів-тварин, плануванням прийомів, веденням медичної документації, обліком послуг і фінансових операцій;

- проведено огляд існуючих CRM-систем загального й спеціалізованого призначення, що дало можливість виявити їхні функціональні та архітектурні обмеження, зокрема недостатню адаптованість до українського ринку та неповну підтримку ветеринарної специфіки;

- сформовано комплекс функціональних і нефункціональних вимог, які стали основою для подальшого технічного та програмного проєктування.

Етап архітектурно-структурного проєктування забезпечив створення чіткої логічної та технологічної структури системи. Зокрема:

- розроблено інфологічну (ER) модель даних, яка визначила сім ключових сутностей системи та встановила зв'язки між ними, забезпечуючи реляційну цілісність та несуперечність інформації;

- побудовано структурні та функціональні діаграми (DFD, UML Use Case), що дали змогу візуалізувати внутрішні процеси CRM-системи, ролі користувачів та основні сценарії взаємодії;

- обґрунтовано вибір трирівневої архітектури (Presentation – Application –

Data), яка забезпечує модульність, масштабованість, захищеність і потенціал для майбутнього розширення функціоналу;

- спроектовано REST-інтерфейси та стандартизовані API-ресурси для підтримки гнучкого обміну даними між рівнями системи.

Етап програмної реалізації та тестування підтвердив працездатність і відповідність системи сформульованим вимогам. Було:

- реалізовано ключові функціональні модулі («Облік клієнтів та пацієнтів», «Календар та прийоми», «Електронна медична картка», «Фінанси та облік послуг», «Комунікації», «Адміністрування»), а також побудовано SPA-інтерфейс користувача із сучасною логікою взаємодії;

- впроваджено механізми автентифікації, авторизації та розмежування доступу (RBAC), що підвищує безпеку системи;

- здійснено тестування за багаторівневою стратегією (Unit, Integration, System Testing), яке засвідчило коректність роботи критичних функцій, відповідність програмних модулів бізнес-логіці та стійкість системи до помилок;

- підготовлено інструкцію користувача та рекомендації щодо впровадження системи у ветеринарному центрі.

Узагальнюючи результати роботи, можна стверджувати, що розроблена CRM-система є технічно обґрунтованим, функціонально завершеним та безпечним програмним продуктом, який повністю відповідає сучасним вимогам до інформаційних систем у сфері медичного та сервісного обслуговування. Запропоноване рішення забезпечує комплексну автоматизацію основних бізнес-процесів ветеринарного центру, підвищує ефективність роботи персоналу, покращує якість взаємодії з клієнтами та створює передумови для подальшого розвитку та масштабування системи.

Перспективи подальших досліджень полягають у розширенні функціональності CRM-системи шляхом інтеграції з лабораторним обладнанням та зовнішніми сервісами, впровадженні мобільних застосунків для персоналу та клієнтів, розробленні аналітичних і прогнозних модулів на основі методів машинного навчання, удосконаленні механізмів безпеки та захисту даних

відповідно до сучасних стандартів, а також у дослідженні можливостей масштабування системи для мережі ветеринарних центрів і впровадженні елементів телемедицини.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

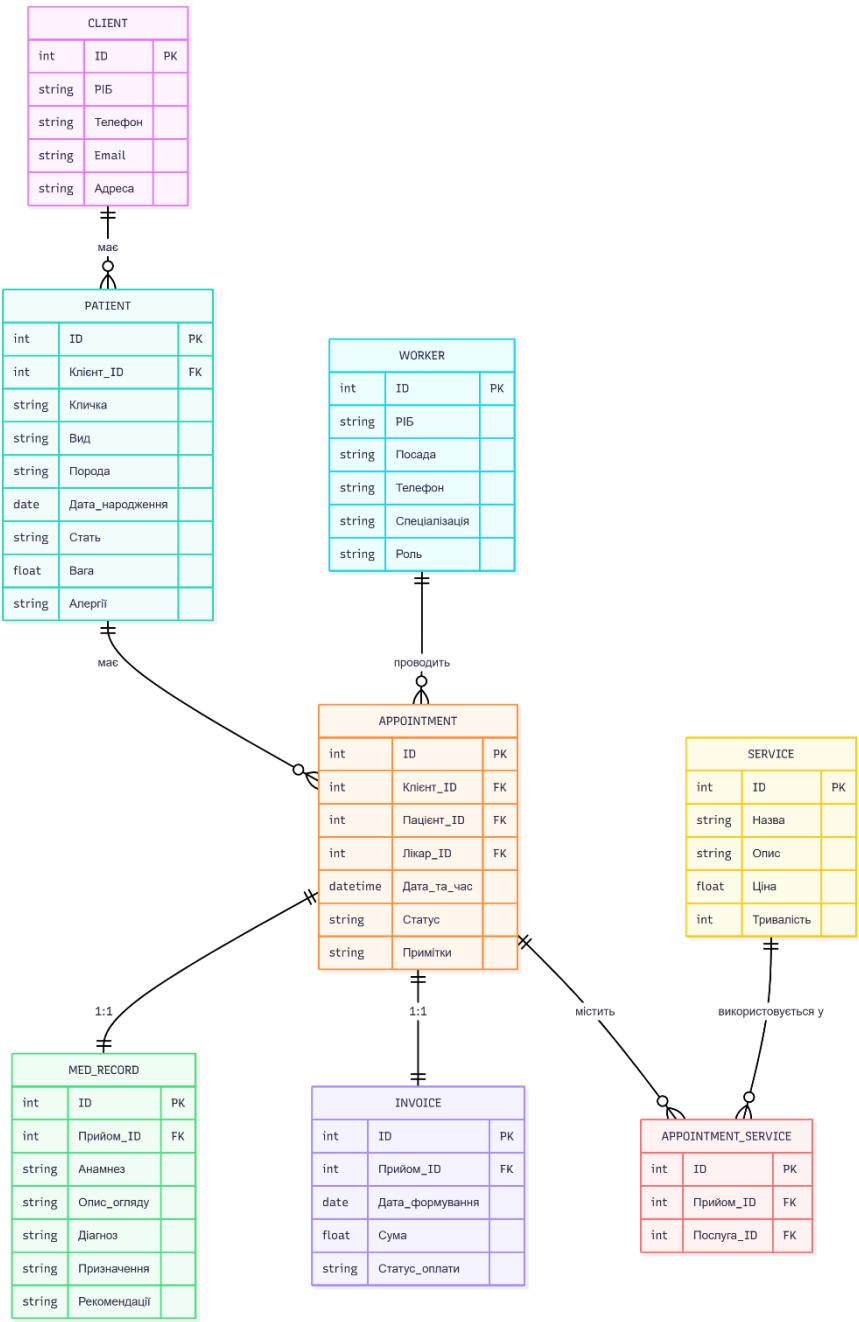
1. Басараба Н. І. Інформаційні системи і технології в охороні здоров'я. Львів: ЛНУ ім. Франка, 2019. 248 с.
2. Літвіненко О. В. Інформаційні системи та бази даних: навч. посіб. Київ: КНЕУ, 2020. 312 с.
3. Корченко О. Г., Федорчук Є. А. Інформаційна безпека: теорія та практика. Київ: ДУТ, 2018. 420 с.
4. Лимаренко В. І. Проектування інформаційних систем. Київ: Кондор, 2021. 286 с.
5. Sheldon R. SQL: The Complete Reference. New York: McGraw-Hill, 2018. 912 p.
6. Elmasri R., Navathe S. Fundamentals of Database Systems. 7th ed. Boston: Pearson, 2017. 1264 p.
7. Date C. J. Database Design and Relational Theory. 2nd ed. Sebastopol: O'Reilly Media, 2019. 272 p.
8. Pressman R., Maxim B. Software Engineering: A Practitioner's Approach. 8th ed. New York: McGraw-Hill, 2020. 1024 p.
9. Sommerville I. Software Engineering. 10th ed. London: Pearson, 2016. 816 p.
10. Fowler M. UML Distilled. 3rd ed. Boston: Addison-Wesley, 2018. 208 p.
11. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 1994. 395 p.
12. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures. University of California, 2000. 183 p.
13. Richardson L., Ruby S. RESTful Web Services. Sebastopol: O'Reilly Media, 2007. 454 p.
14. Martin R. C. Clean Code. Upper Saddle River: Prentice Hall, 2008. 464 p.
15. Martin R. C. Clean Architecture. Boston: Prentice Hall, 2017. 432 p.
16. International Organization for Standardization. ISO/IEC 27001:2022 Information security management systems. Geneva: ISO, 2022.

17. Nielsen J. Usability Engineering. San Diego: Morgan Kaufmann, 1993. 362 p.
18. Krug S. Don't Make Me Think. 3rd ed. Berkeley: New Riders, 2014. 216 p.
19. Postman. Postman API Platform Documentation. 2023. URL: <https://learning.postman.com/> (дата звернення: 15.02.2025).
20. Google. Material Design Guidelines. 2023. URL: <https://m3.material.io> (дата звернення: 15.02.2025).
21. Mozilla Developer Network. JavaScript Documentation. 2023. URL: <https://developer.mozilla.org> (дата звернення: 15.02.2025).
22. Django REST Framework. Official documentation. 2023. URL: <https://www.django-rest-framework.org/> (дата звернення: 15.02.2025).
23. Meta Platforms. React Documentation. 2023. URL: <https://react.dev> (дата звернення: 15.02.2025).
24. PostgreSQL Global Development Group. PostgreSQL Documentation. 2023. URL: <https://www.postgresql.org/docs/> (дата звернення: 15.02.2025).
25. VetCRM — Veterinary Clinic Management System. 2023. URL: <https://vetcrm.net> (дата звернення: 15.02.2025).
26. IDEXX. eVetPractice Documentation. 2023. URL: <https://idexx.com> (дата звернення: 15.02.2025).
27. Provet Cloud. Official documentation. 2023. URL: <https://www.provet.cloud> (дата звернення: 15.02.2025).
28. Helsi.me. Інформаційна медична система. 2024. URL: <https://helsi.me> (дата звернення: 15.02.2025).
29. Doctor Eleks. Медична інформаційна система. 2023. URL: <https://doctoreleks.com> (дата звернення: 15.02.2025).
30. Salesforce. Health Cloud Documentation. 2023. URL: <https://www.salesforce.com/> (дата звернення: 15.02.2025).
31. Zoho Corporation. Zoho CRM for Healthcare. 2023. URL: <https://www.zoho.com/> (дата звернення: 15.02.2025).

32. Буров Є. В., Королюк О. М. Архітектура інформаційних систем. Київ: НАУ, 2020. 256 с.
33. Бойко І. М. Системи управління базами даних: навчальний посібник. Харків: ХНЕУ, 2019. 284 с.
34. Турчин В. А. Інформаційні технології в менеджменті. Тернопіль: ТНТУ, 2020. 268 с.
35. Глушко А. В. Веб-програмування: навчальний посібник. Київ: КНУ, 2021. 304 с.
36. Ковальчук Н. А. Основи моделювання інформаційних систем. Львів: Новий Світ, 2018. 218 с.
37. International Organization for Standardization. ISO/IEC 25010:2011 Systems and software quality requirements and evaluation. Geneva: ISO, 2011.
38. Swagger. OpenAPI Specification. 2023. URL: <https://swagger.io/specification/> (дата звернення: 15.02.2025).
39. World Wide Web Consortium. Architecture of the World Wide Web. W3C Recommendation, 2004. URL: <https://www.w3.org/TR/webarch/> (дата звернення: 15.02.2025).
40. Cooper A., Reimann R., Cronin D. About Face: The Essentials of Interaction Design. 4th ed. Indianapolis: Wiley, 2014. 720 p.
41. Lewis J., Fowler M. Microservices: a definition of this new architectural term. ThoughtWorks, 2014. URL: <https://martinfowler.com/articles/microservices.html> (дата звернення: 15.02.2025).
42. Kurose J., Ross K. Computer Networking: A Top-Down Approach. 8th ed. Boston: Pearson, 2021. 864 p.
43. Шматков С. О. Інженерія програмного забезпечення: методи, моделі, технології. Харків: ХНУРЕ, 2022. 372 с.
44. Кузнецова Л. А. Технології розробки веб-додатків. Одеса: ОНПУ, 2019. 256 с.
45. European Medicines Agency. Veterinary Medicines Information. 2023. URL: <https://www.ema.europa.eu> (дата звернення: 15.02.2025).

ДОДАТКИ

Додаток А. ER-діаграма інфологічної моделі CRM-системи ветеринарного центру



Додаток Б. Діаграми системи

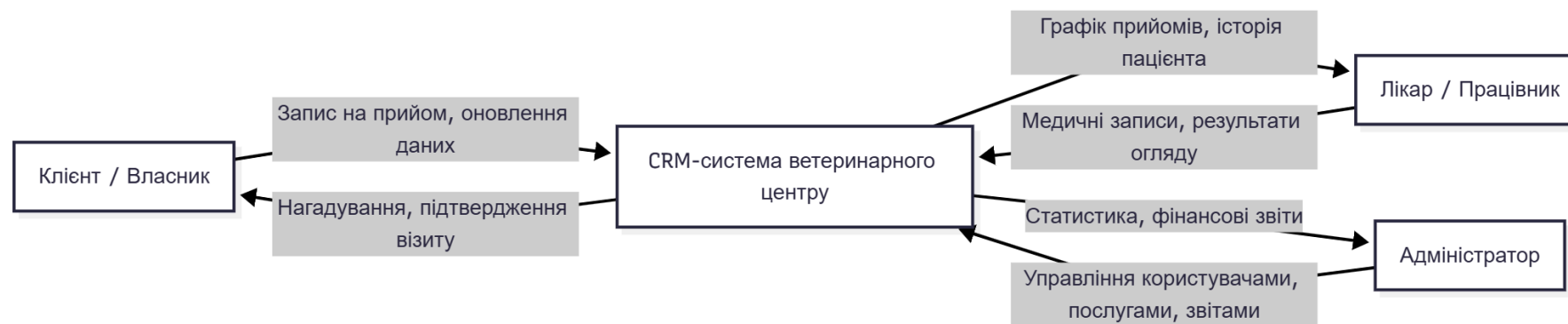


Рис. Б. 1. контекстна діаграма CRM-системи ветеринарного центру

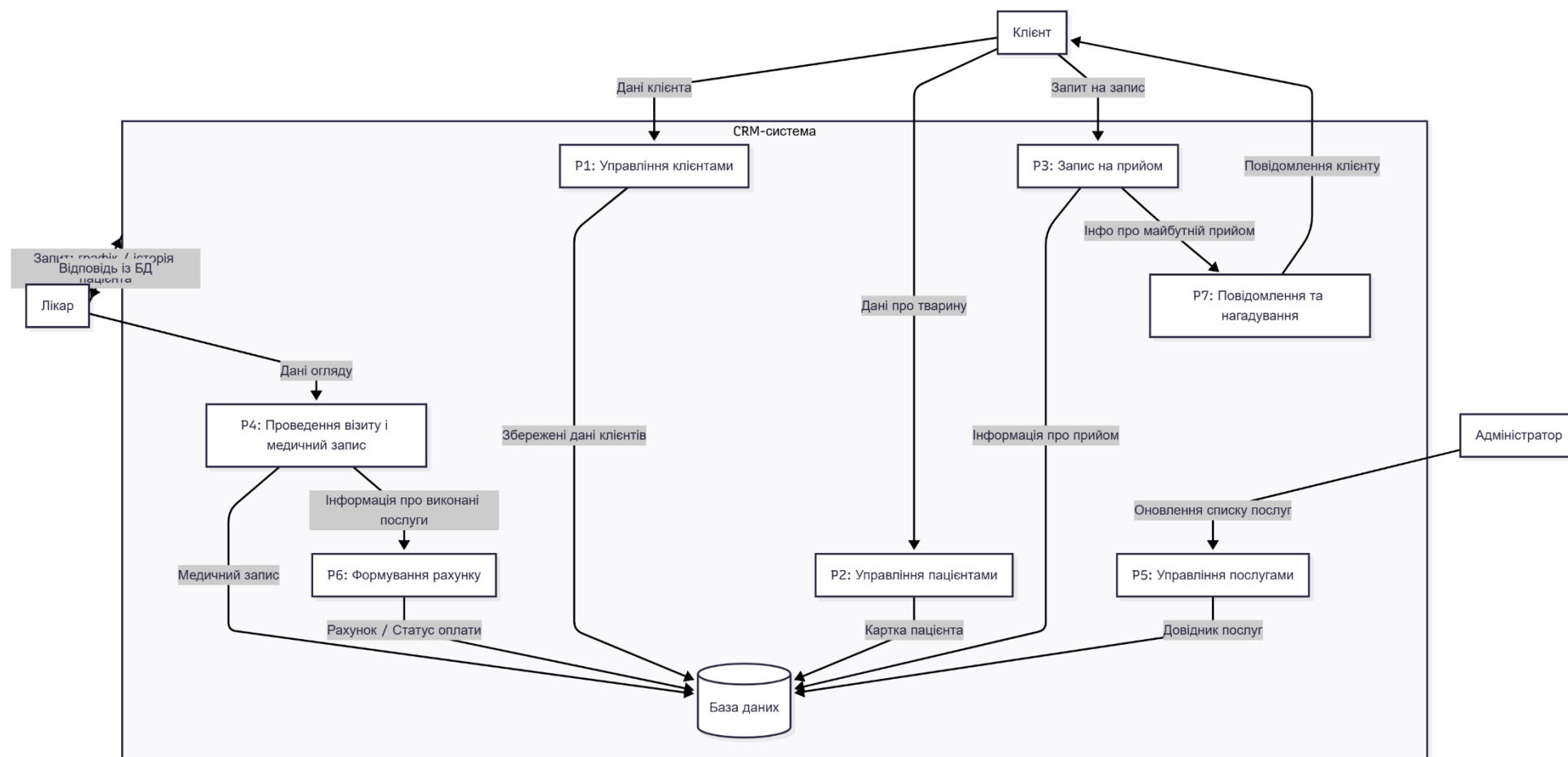


Рис. Б. 2. Деталізація внутрішніх процесів CRM-системи ветеринарного центру

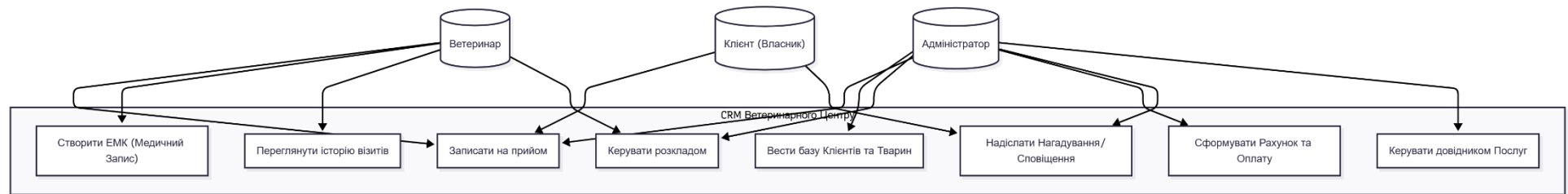


Рис. Б. 3. Діаграма варіантів використання CRM-системи ветеринарного центру

Додаток В. Інтерфейс системи

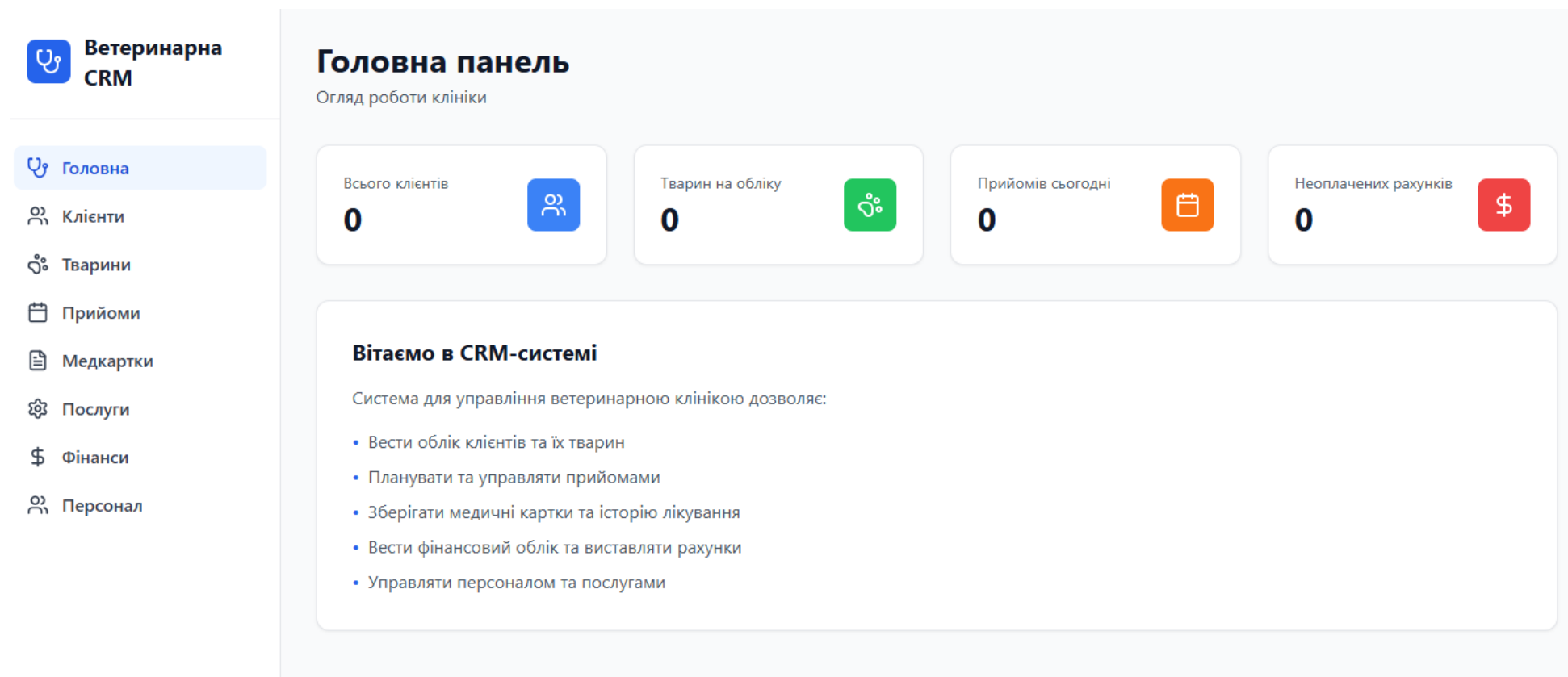


Рис. В.1.Макет головної панелі (Dashboard) CRM-системи

The image shows a web application for a veterinary CRM. On the left is a sidebar with navigation links: Головна, Клієнти (highlighted), Тварини, Прийоми, Медкартки, Послуги, Фінанси, and Персонал. The main header area includes the logo 'Ветеринарна CRM' and the title 'Клієнти' with the subtitle 'Управління базою клієнтів'. A '+ Новий клієнт' button is in the top right. A modal window titled 'Новий клієнт' is open in the center, containing a form with the following fields: 'ПІБ *' (filled with 'Іванов Іван Іванович'), 'Телефон *' (filled with '+380501234567'), 'Email' (filled with 'email@example.com'), 'Адреса' (filled with 'вул. Шевченка, 1'), and 'Примітки' (with placeholder text 'Додаткова інформація...'). At the bottom of the modal are two buttons: 'Зберегти' (blue) and 'Скасувати' (gray). The bottom left of the sidebar shows the user role 'Адміністратор'.

Ветеринарна CRM

Клієнти

Управління базою клієнтів

+ Новий клієнт

Головна

Клієнти

Тварини

Прийоми

Медкартки

Послуги

Фінанси

Персонал

Адміністратор

Новий клієнт

ПІБ *

Іванов Іван Іванович

Телефон *

+380501234567

Email

email@example.com

Адреса

вул. Шевченка, 1

Примітки

Додаткова інформація...

Зберегти

Скасувати

Рис. В.2.Макет інтерфейсу для додавання нового клієнта

Ветеринарна CRM

Тварини

Облік та медіа

Головна

Клієнти

Тварини

Прийоми

Медкартки

Послуги

Фінанси

Персонал

Адміністратор

Адміністратор

Додати тварину

Власник *

Оберіть клієнта

Кличка *

Рекс

Вид *

Собака, Кіт, Папуга...

Порода

Німецька вівчарка

Стать

Невідома

Дата народження

ДД.ММ.ГГГГ

Вага (кг)

5.5

Колір

Чорний

Мікрочіп

123456789

Алергії

Алергія1, Алергія2 (розділяйте комами)

Примітки

Додаткова інформація...

Зберегти

Скасувати

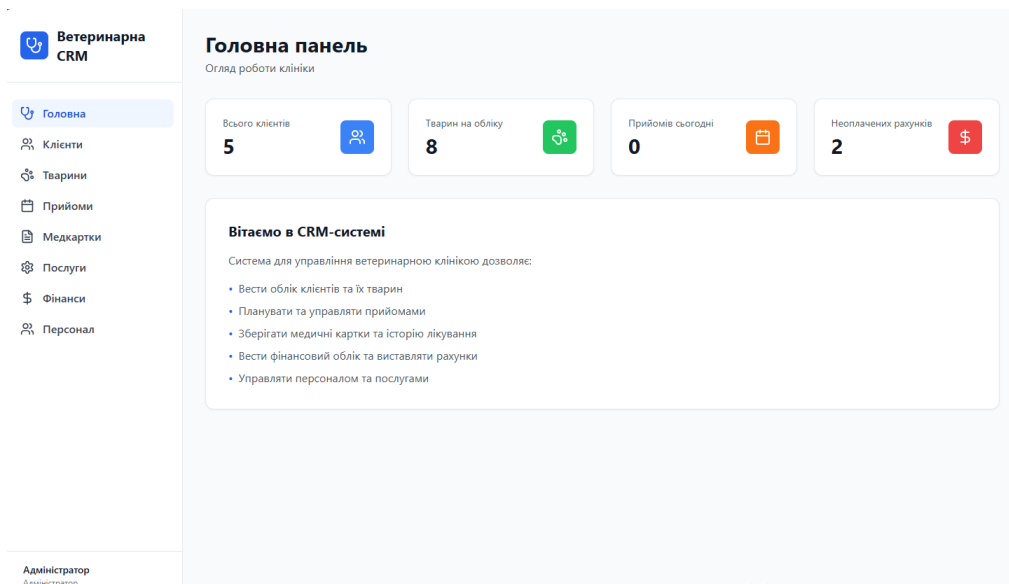
+ Додати тварину

Рис. В.3.Макет інтерфейсу для додавання тварини (пацієнта)

Додаток Г. Керівництво користувача та навчання персоналу

А. Вступ та загальний огляд інтерфейсу

- **Опис структури:** Представляється опис загальної структури системи, включаючи **Головну панель (Dashboard)** (Рис. Г.1), що слугує єдиною точкою входу та відображає ключові метрики (наприклад, кількість запланованих прийомів).



- **Принципи навігації:** Пояснюється робота **Сайдбару** та його відповідність функціональним модулям (Таблиця 2.2), а також використання фільтрів та функцій пошуку для швидкого доступу до даних.

Б. Розмежування доступу (RBAC)

- **Пояснення ролей:** Чітко пояснюється, які саме модулі та функції доступні для кожної ролі користувача (Адміністратор, Ветеринар, Клієнт). Посилання робиться на **Діаграму варіантів використання UML** (Додаток В, Рис. В.3), яка є візуальним підтвердженням **Розмежування прав доступу (RBAC)**.
 - *Приклад:* Ветеринар має право доступу до **Модуля ЕМК** на запис, тоді як Адміністратор має лише право перегляду.
- **Адміністратор використовує Edge Function setup-users** для створення облікових записів персоналу. Адміністратор видає email та пароль новим співробітникам. Клієнти не мають доступу до системи, їх просто додає персонал через форму

В. Покрокові інструкції за модулями

Надається детальний опис робочих процесів (**алгоритмів взаємодії**) для кожного функціонального блоку:

1. Модуль «Клієнти та Пацієнти»:

- **Інструкція з введення даних:** Покрокове заповнення форми «Новий клієнт» (Рис. Г.2) та форми «Додати Пацієнта» (Рис. Г.3), із наголосом на обов'язковості заповнення полів зовнішніх ключів (Клієнт_ID).

- - **Операції:** Процедури пошуку за критеріями (ПІБ, телефон, номер карти) та редагування профілів.
- 2. **Модуль «Календар»:**
 - **Процедура Запису:** Алгоритм перевірки доступності лікаря (Р3) та фіксації прийому.
 - **Обробка змін:** Інструкції з перенесення (зміна часу/лікаря) та скасування візиту із фіксацією причини зміни статусу.
- 3. **Модуль «ЕМК»:**
 - **Створення Запису (Р4):** Детальний опис заповнення структурованих полів Медичного Запису (Анамнез, Огляд, Діагноз, Призначення).
 - **Робота з файлами:** Процедура прикріплення зовнішніх файлів (результати аналізів, знімки), що зберігаються на сервері та пов'язуються з конкретним Прийом_ID.
- 4. **Управління фінансами:**
 - **Формування рахунку (Р5):** Порядок генерації INVOICE на основі обраних послуг та механізм перевірки агрегованої суми.
 - **Фіксація оплат (Р6):** Реєстрація факту оплати та оновлення статусу Рахунку.

Програма практичного навчання персоналу

Для забезпечення швидкого переходу на нову систему проводиться **практичне навчання**, сфокусоване на відпрацюванні ключових робочих процесів.

Роль	Ключовий фокус навчання	Критичні навички, що відпрацьовуються
Адміністратор	Операційна ефективність та Календарний менеджмент	Реєстрація нових Клієнтів та Пацієнтів (Р1, Р2); управління довідниками (Послуги, Працівники); швидке вирішення конфліктів розкладу (Р3) ; первинний фінансовий облік.

Ветеринар	Медична документація та клінічний потік	Швидкий доступ до ЕМК та історії пацієнта; коректне внесення медичних записів та діагнозів (Р4); правильне формування переліку наданих послуг для фінансового обліку.
------------------	--	--

Практичне навчання включає моделювання реальних сценаріїв, таких як обробка "неявки" клієнта або виправлення помилки у медичному записі. Це забезпечує не просто знання функцій, а **набуття навичок** ефективного використання системи у динамічних умовах ветеринарної клініки.

Для тестування системи передбачені наступні облікові записи:

admin@clinic.com / admin

doctor1@clinic.com / doctor

client1@clinic.com / client1